Distributed Network Algorithms

Welcome to the GIAN Course!

Stefan Schmid
Aalborg University, Denmark
& TU Berlin, Germany

Partha Sarathi Mandal
IIT Guwahati, India

Goal: Get a feeling of the field and techniques. Essentially: Classic TCS problems from a distributed lens, also an active research area.

Distributed Network Algorithms

Welcome to the GIAN Course!

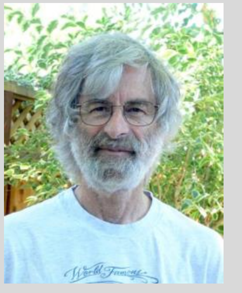
Here the next 10 days. Let‘s meet! ☺

Stefan Schmid
Aalborg University, Denmark
& TU Berlin, Germany

Partha Sarathi Mandal
IIT Guwahati, India

What is a distributed system? A quote.

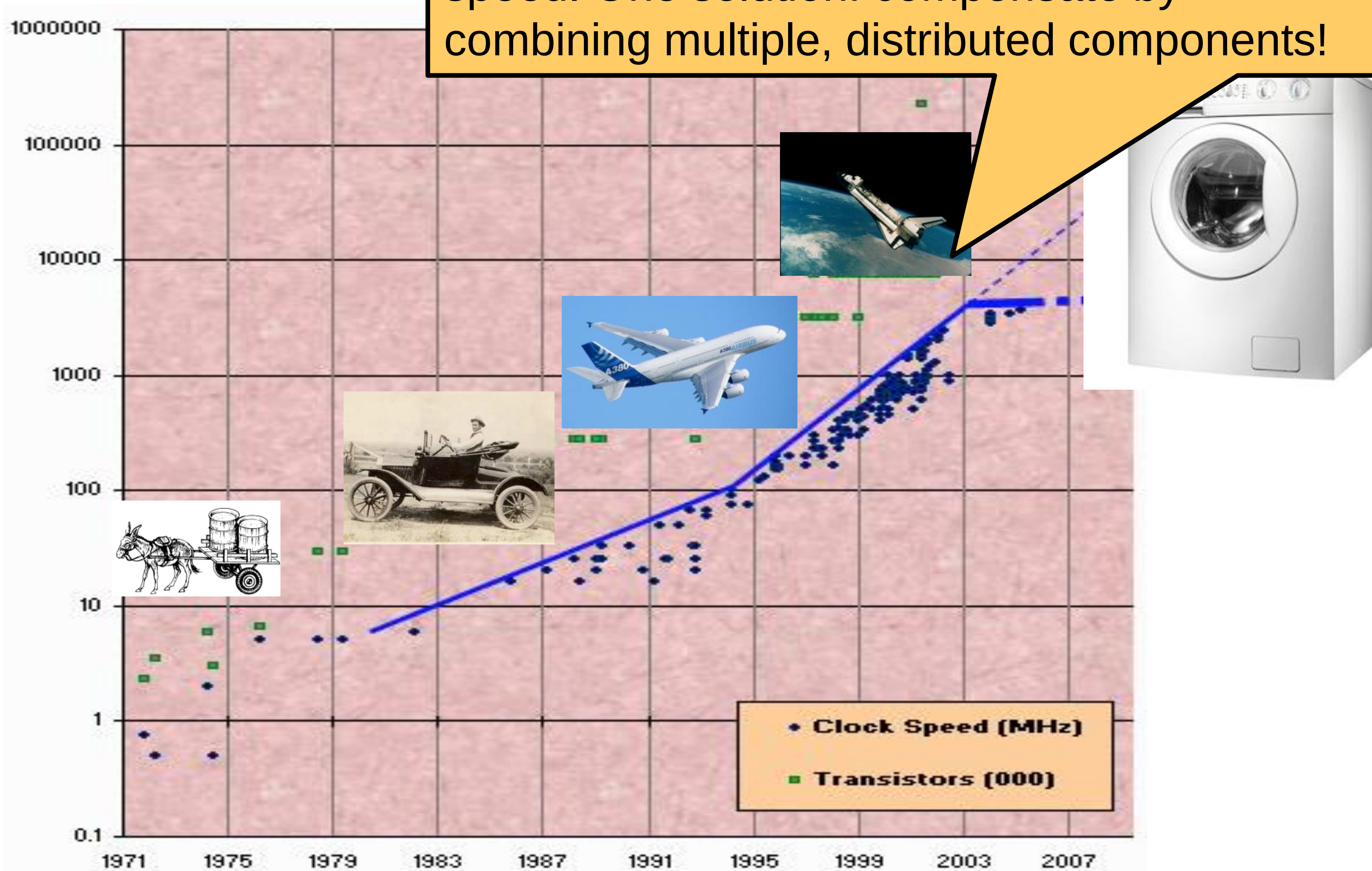
“You know you have a distributed system when the crash of a computer you’ve never heard of stops you from getting any work done.” (Leslie Lamport)



Why distributed systems

Heard of Moore's law?

We start observing the limits of extent at which we can increase transistor density and clock speed. One solution: compensate by combining multiple, distributed components!



Another reason: Big Data

Science

- Traditionally: **astronomers** and **physicists**
  - CERN LHC: 15 petabytes / year
- More recently: **biologists** and **life scientists** join big data club
  - Everything from '*why coastal algae bloom to what microbed dwell where in the human body*'

Roughly 1 TB/day!

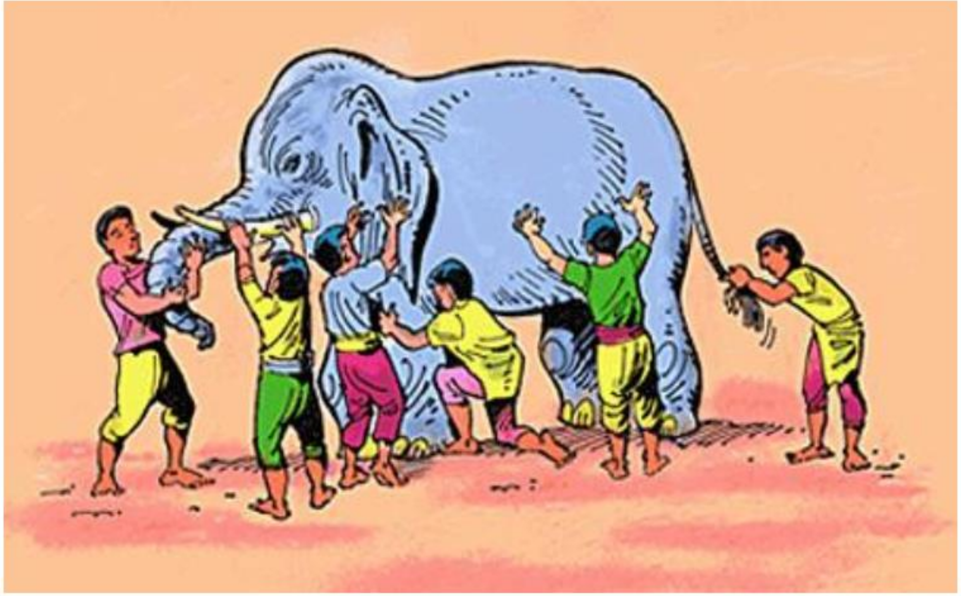
Big data is relevant not only for researchers, but also for businesses, decision makers, journalists, crowd workers, and "everyday consumers".

Business

- Facebook, Twitter, LinkedIn
  - Operate pipelines of 100s terabytes / day
  - Logging user interactions, monitoring compute infrastructures, tracking business-critical functions
- More and more **machine-driven** data collection
- Advent of **Internet-of-Things**, smart devices, …

Processing this data requires distributed systems and algorithms!

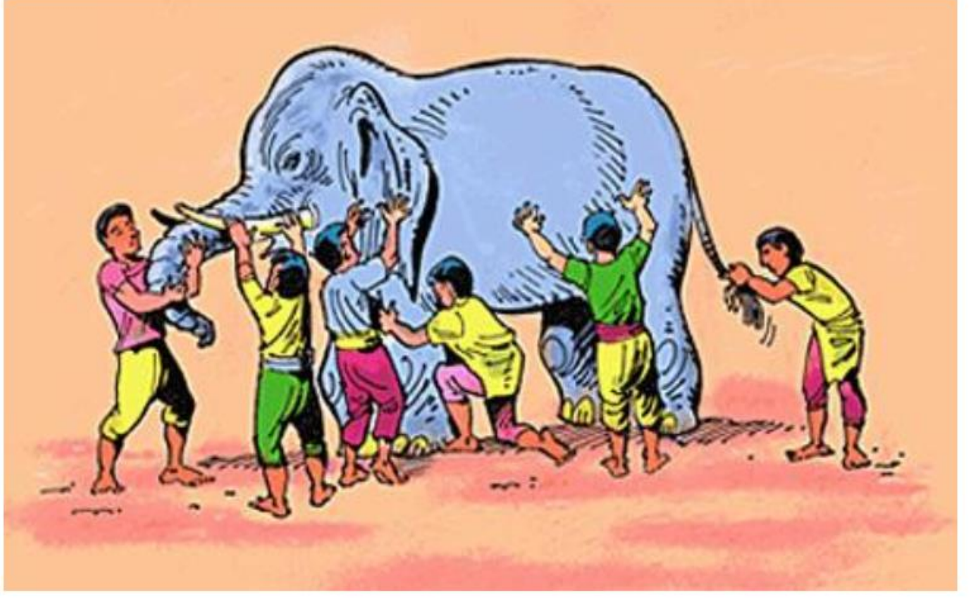
What makes distributed systems challenging? An analogy.



Distributed systems are large beasts: and each individual or constituting component can often only see a **small part**.

Often **hard or impossible** to obtain a global **snapshot**!

What makes distributed systems challenging? An analogy.

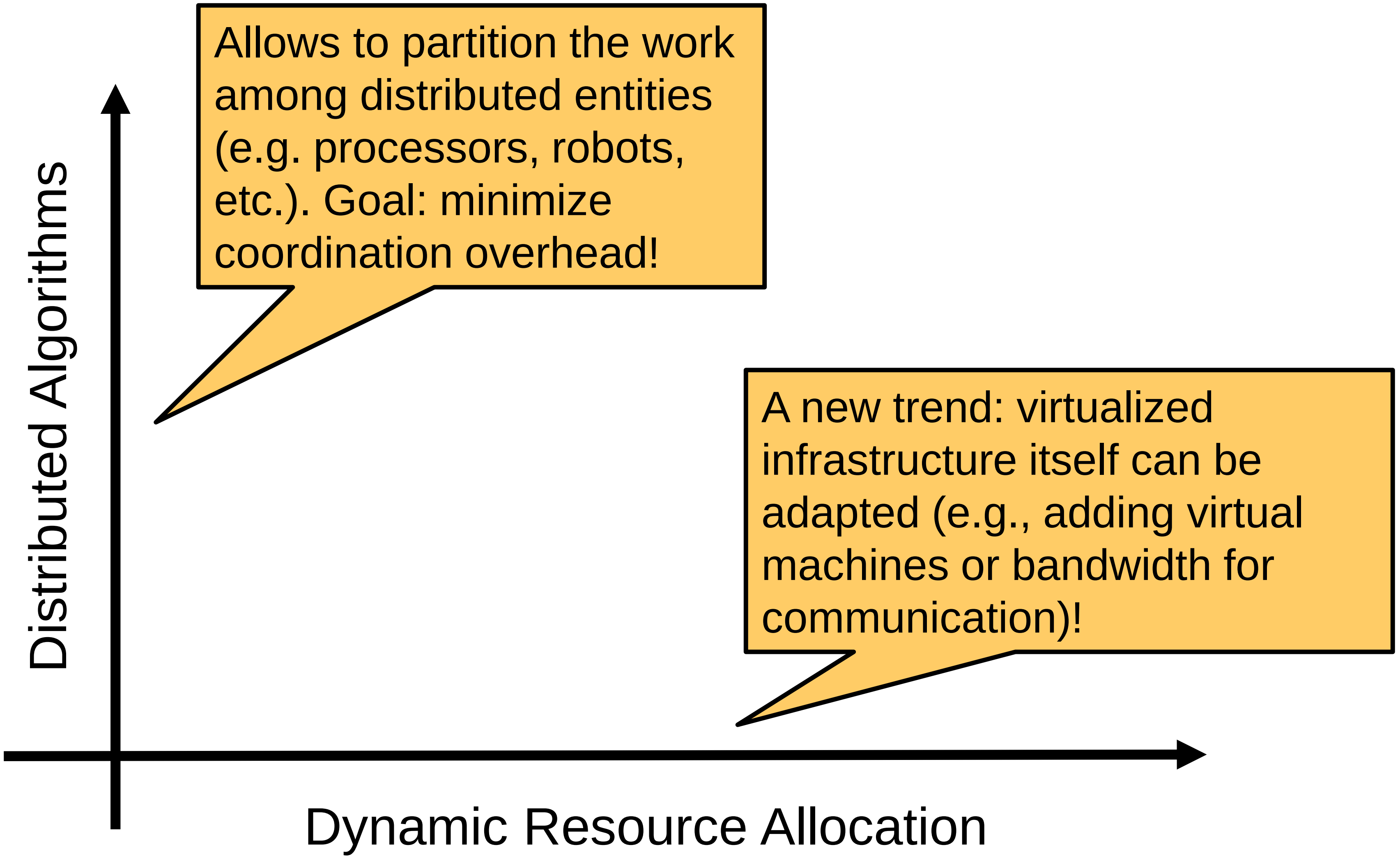


Distributed systems are large beasts: and each individual or constituting component can often only see a **small part**.

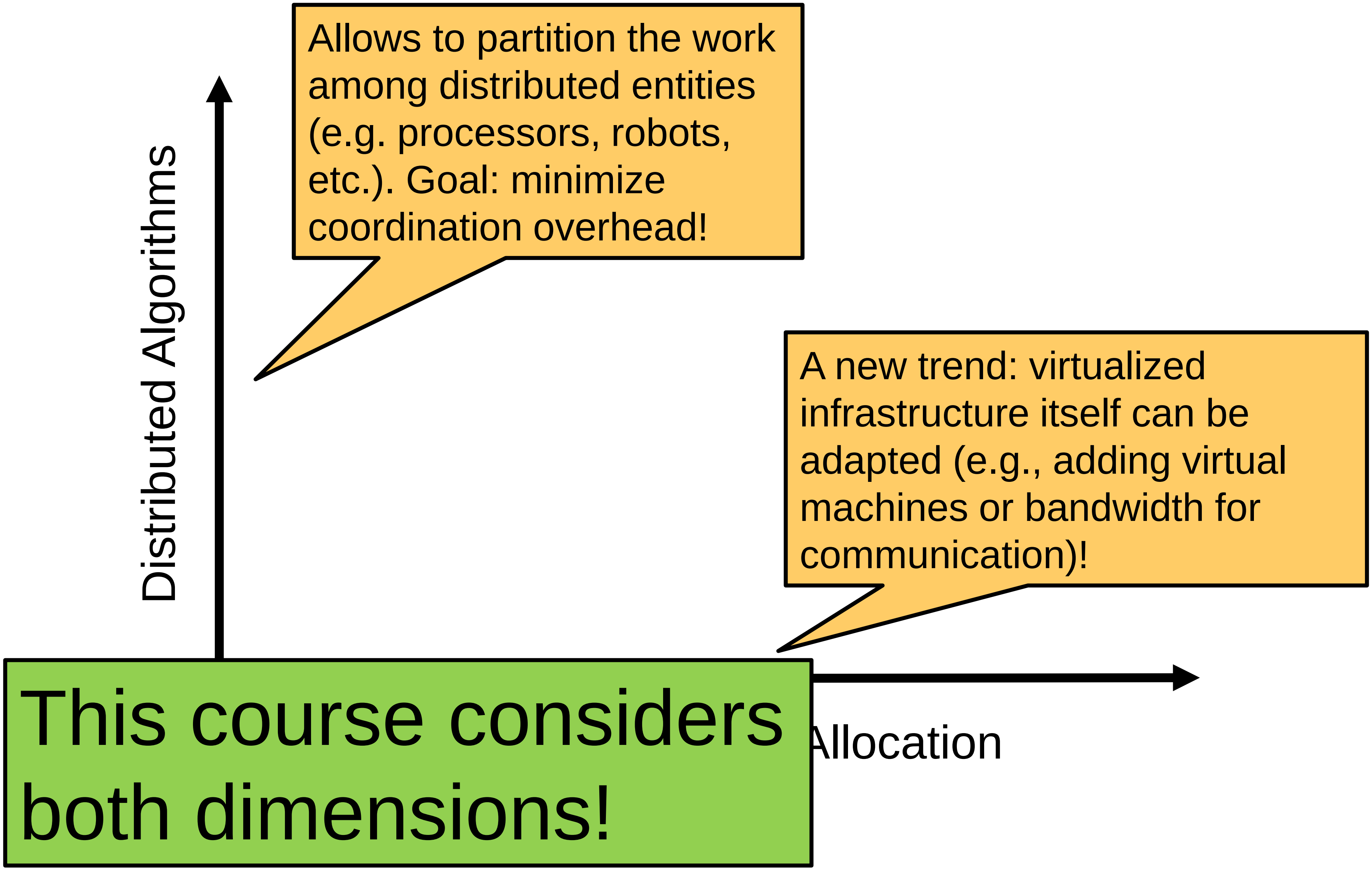
Often **hard or impossible** to obtain a global **snapshot**!

- Human mind not used to think about concurrent processes:
  In my opinion: better to go from theory to practice than vice versa!

How to perform computations in a distributed system?



How to perform computations in a distributed system?



Modern Distributed Systems: Virtualized & Programmable

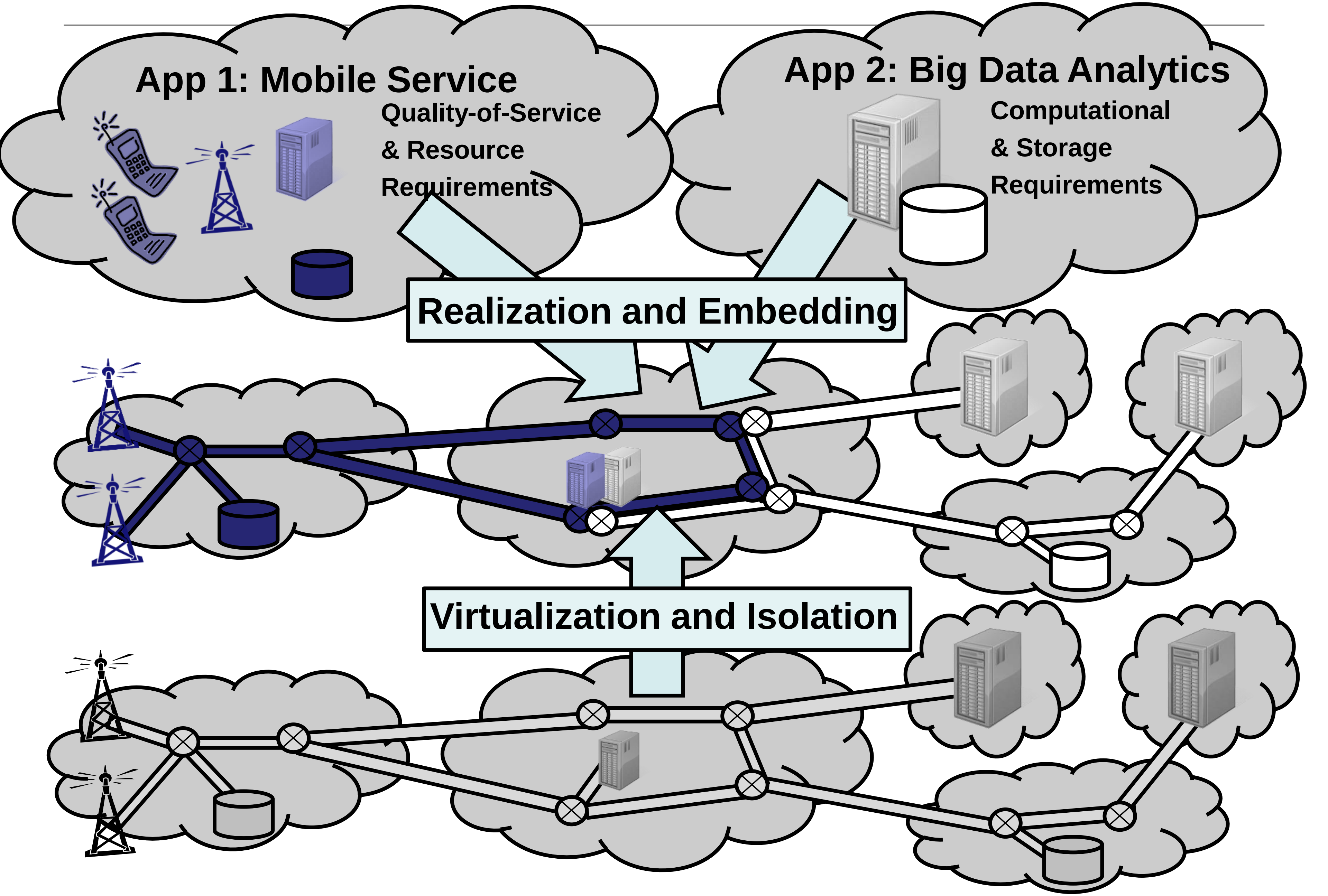
"We are at interesting inflection point!"

Keynote by George Varghese

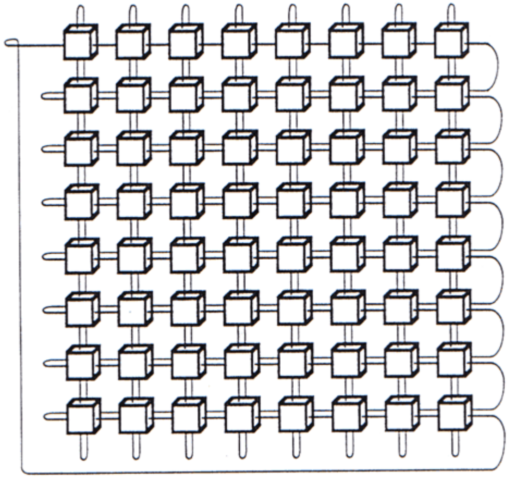
Modern Distributed Systems: Virtualized & Programmable

"We are at interesting inflection point!"
Keynote by George Varghese

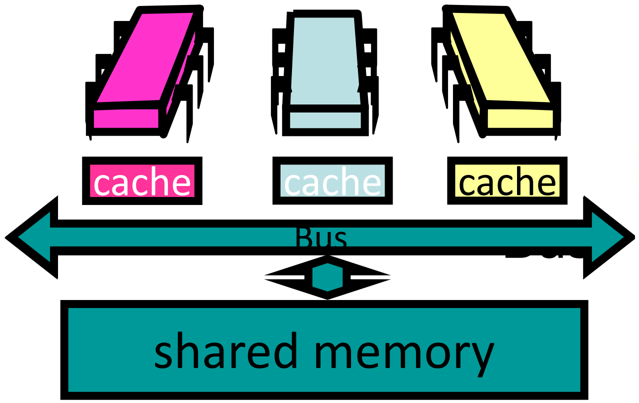
Predictable Performance in Virtualized Environments?



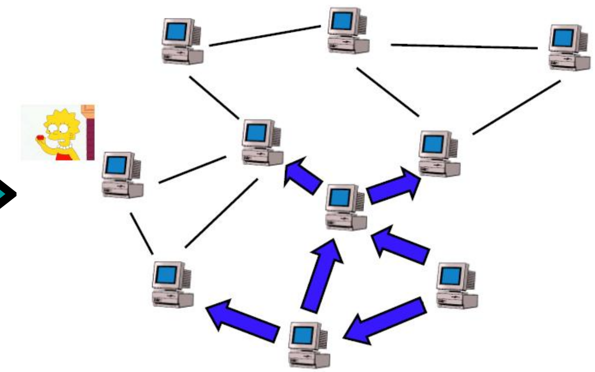
Spectrum of Distributed (Computer) Systems



E.g., tiny graphical processing units (GPUs) and specialized devices, in which large arrays of **simple** processors work in lock-step, PRAM, ...



Multi-threaded + multi-core servers/desktops with shared memory for communication.

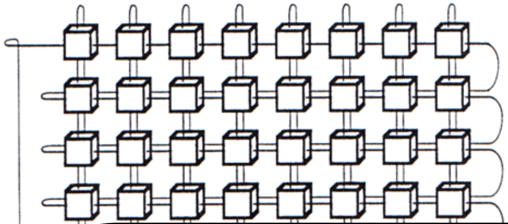


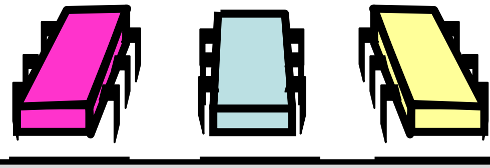
Loosely-coupled **peer-to-peer** systems with message passing communication

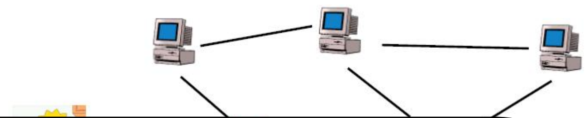
small/synchronous/... ⟷ wide-area/decoupled/...

Even your laptop is a distributed system (multiple cores etc.)!

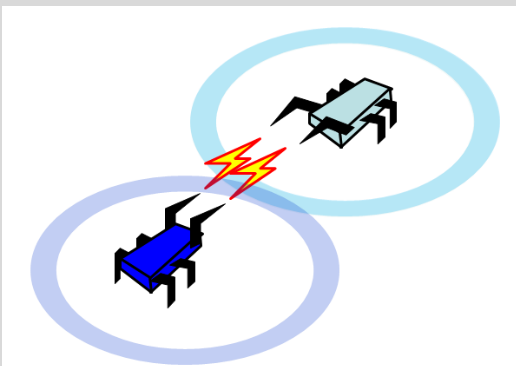
Spectrum of Distributed (Computer) Systems

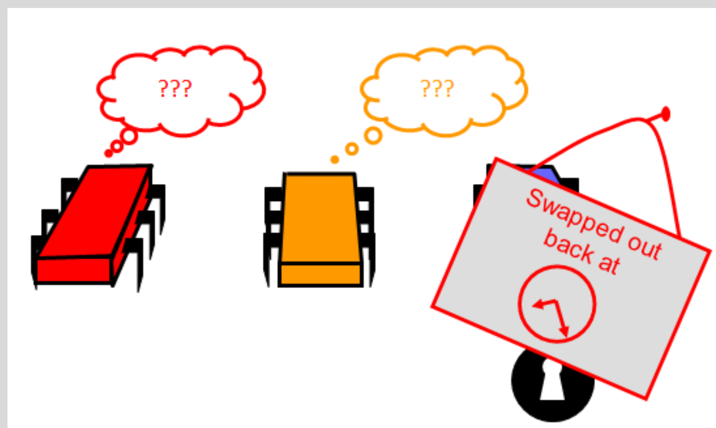


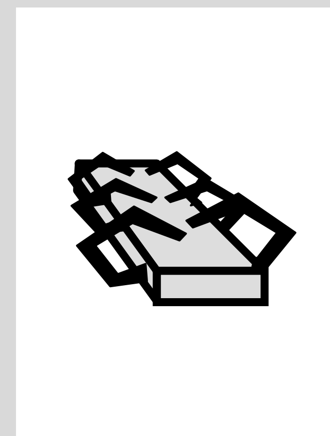




Many commonalities! Communication, concurrent, failures, …







Not a Panacea: Limitations of Distributed Computing

- Some problems can be very hard in distributed setting (e.g., symmetry breaking)
- And not everything is embarrassingly parallel: inherent speedup limits

You can only speedup the parallel work share!

Amdahl's Law

$$S = \frac{1}{\underbrace{1-p}_{\text{sequential part}} + \underbrace{p/n}_{\text{parallel part}}}$$

S = speedup

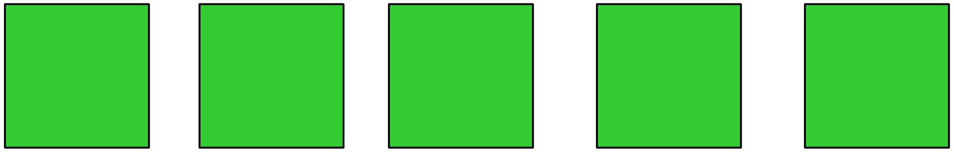
p = fraction of work that can be done in parallel

n = number of processors

Take-home message:
try to minimize sequential part!

Speedup Limits

Example: 5 friends want to paint new apartment, with 5 rooms

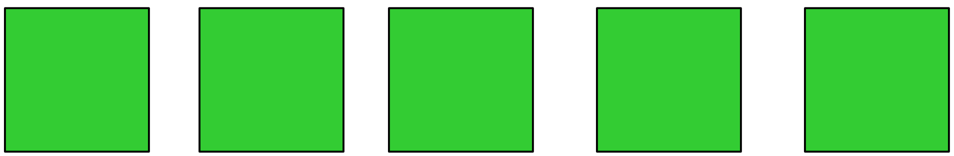


Speed-up 5 people vs 1?

5 times faster than alone!

Speedup Limits

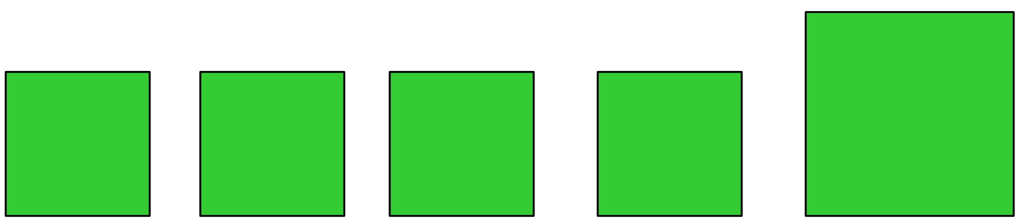
Example: 5 friends want to paint new apartment, with 5 rooms



Speed-up 5 people vs 1?

5 times faster than alone!

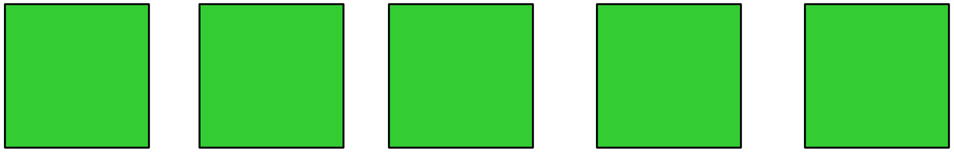
What if last room is twice as large?



- Assume there is space for at most one painter per room!

Speedup Limits

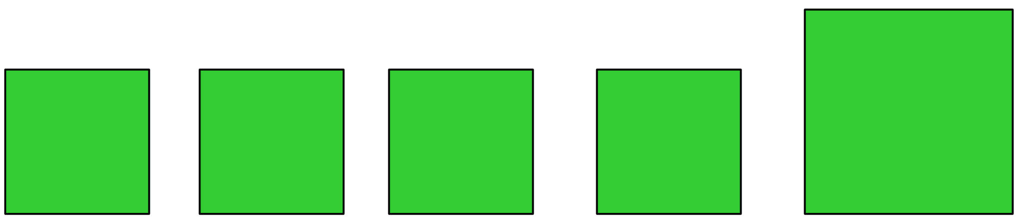
Example: 5 friends want to paint new apartment, with 5 rooms



Speed-up 5 people vs 1?

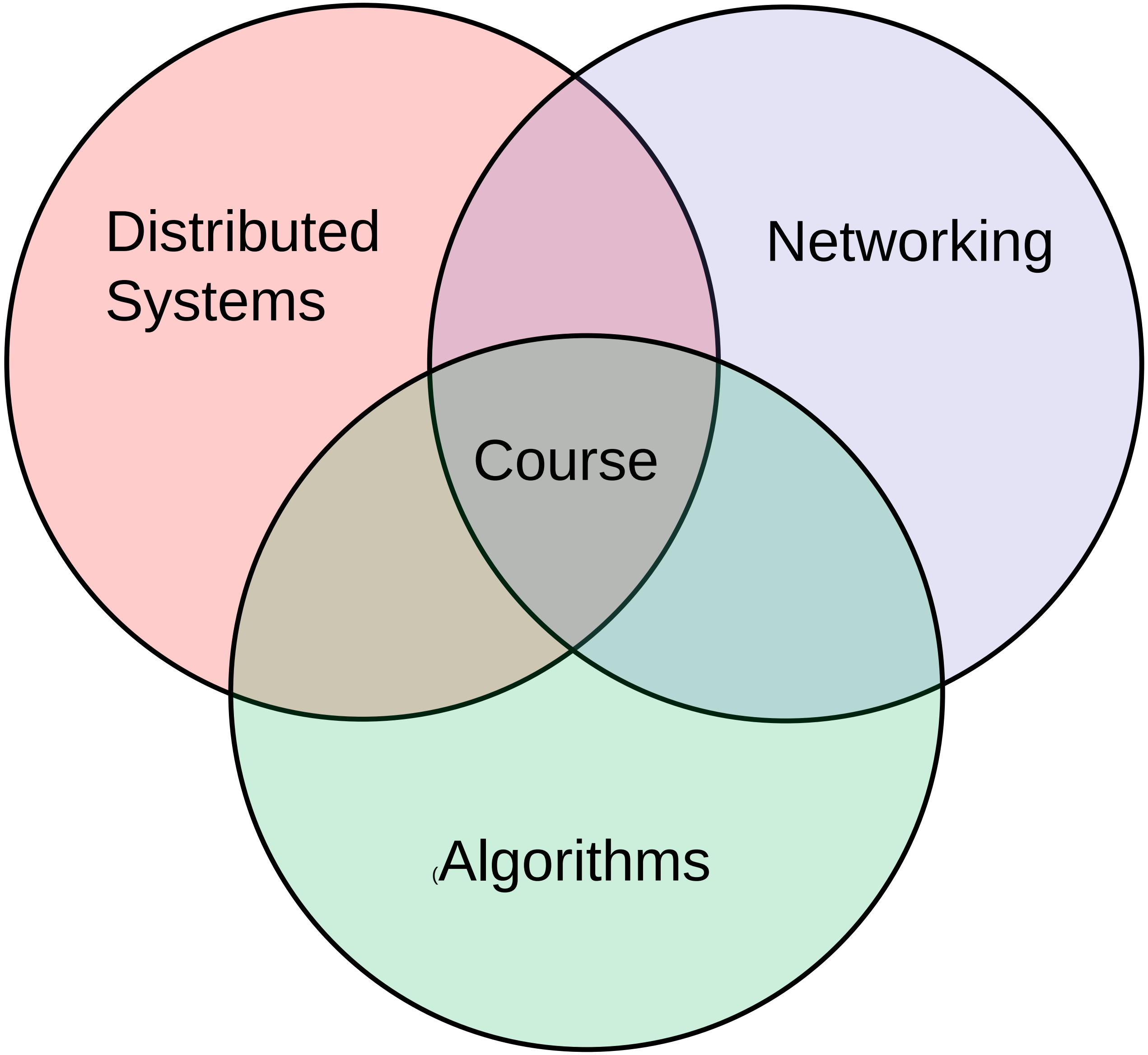
5 times faster than alone!

What if last room is twice as large?

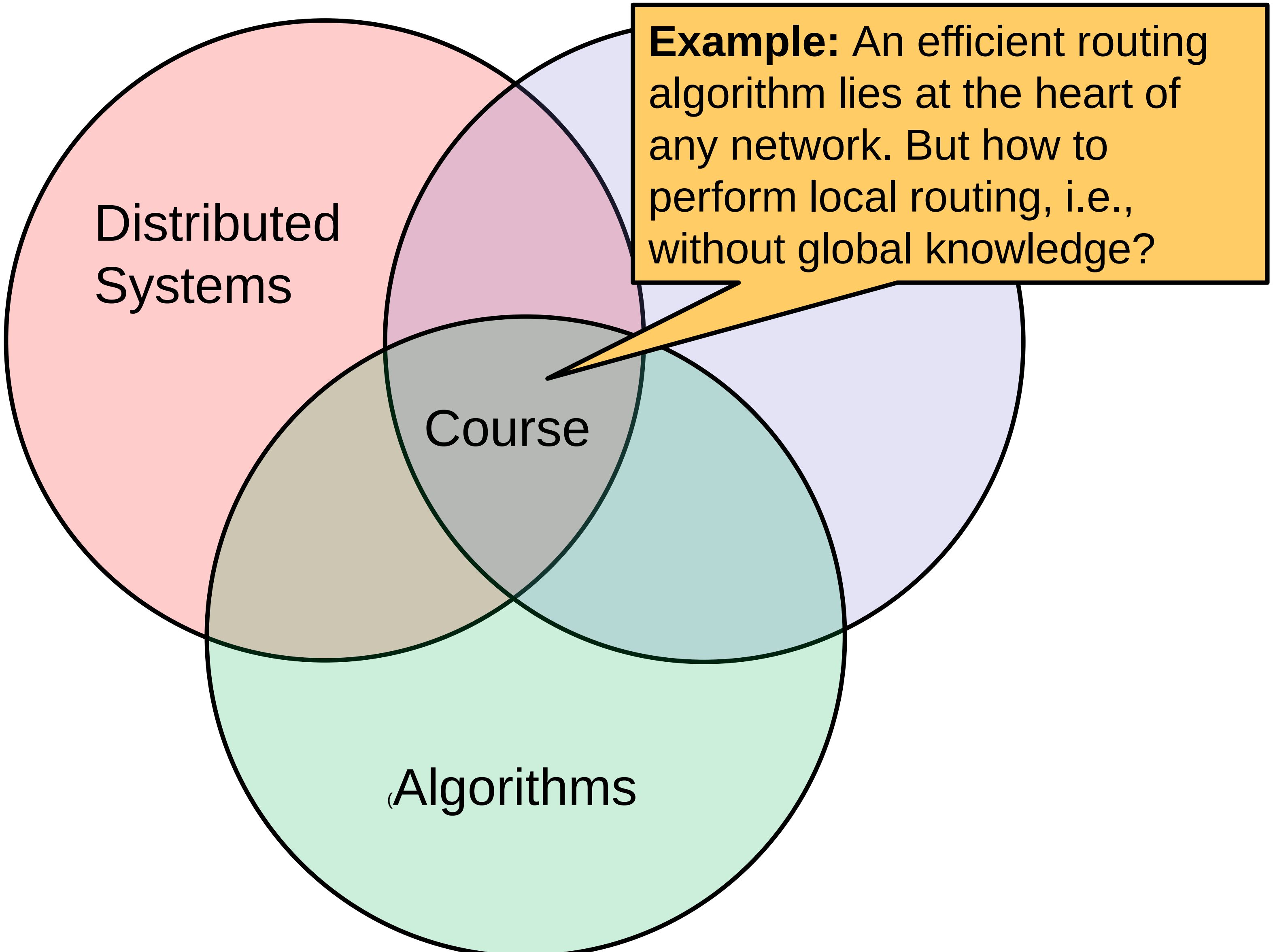


- Assume there is space for at most one painter per room!
- Only $p=5/6$ of the work can be performed in parallel, rest sequential.
- Parallel execution time = $5/(5\cdot 6)+1/6 = 1/6+1/6 = 2/6 = 1/3$. Only 3 times faster!
- Would be better to parallelize painting of last room also!

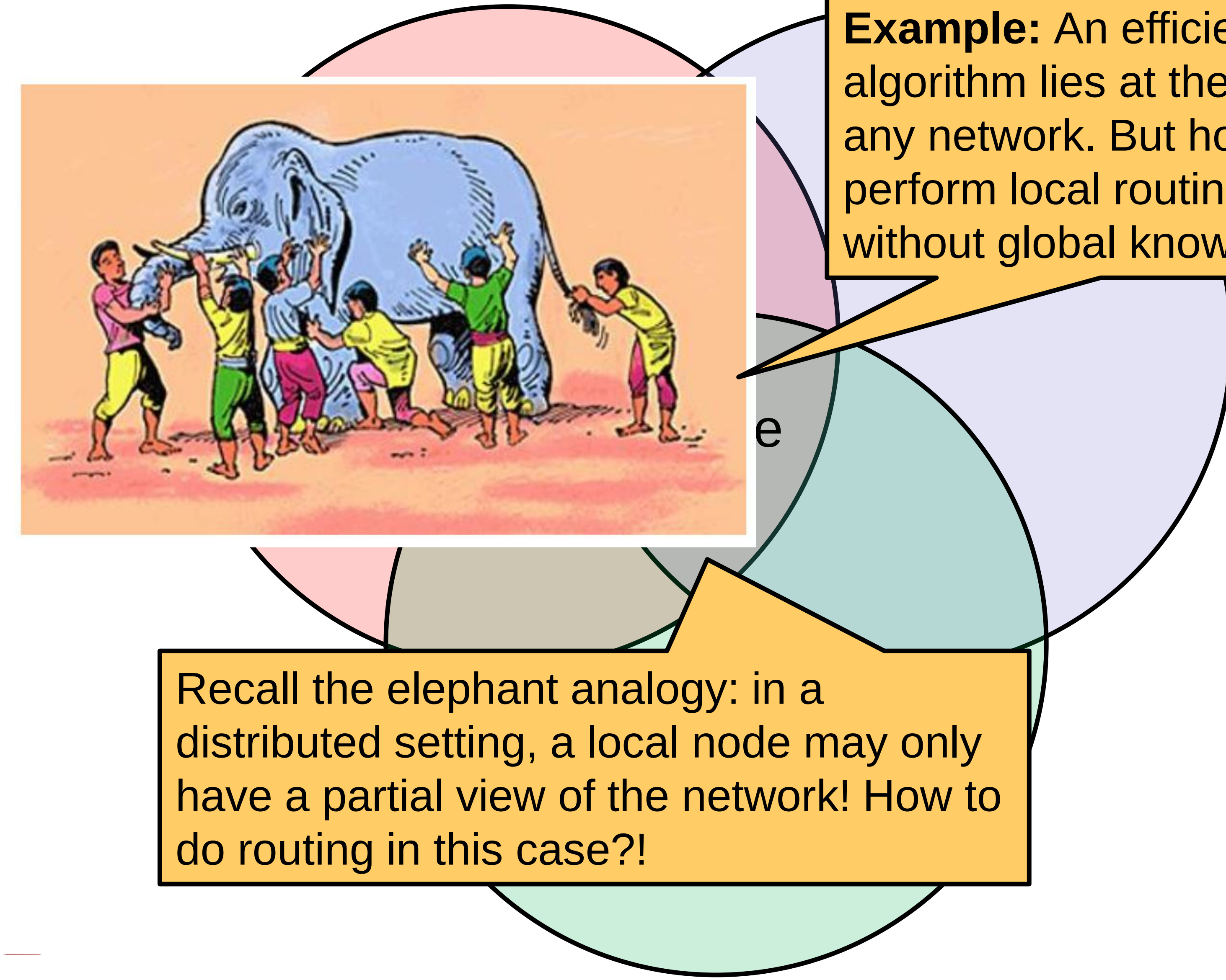
This Course



This Course


Example: An efficient routing algorithm lies at the heart of any network. But how to perform local routing, i.e., without global knowledge?
Distributed Systems
Course
Algorithms

This Course

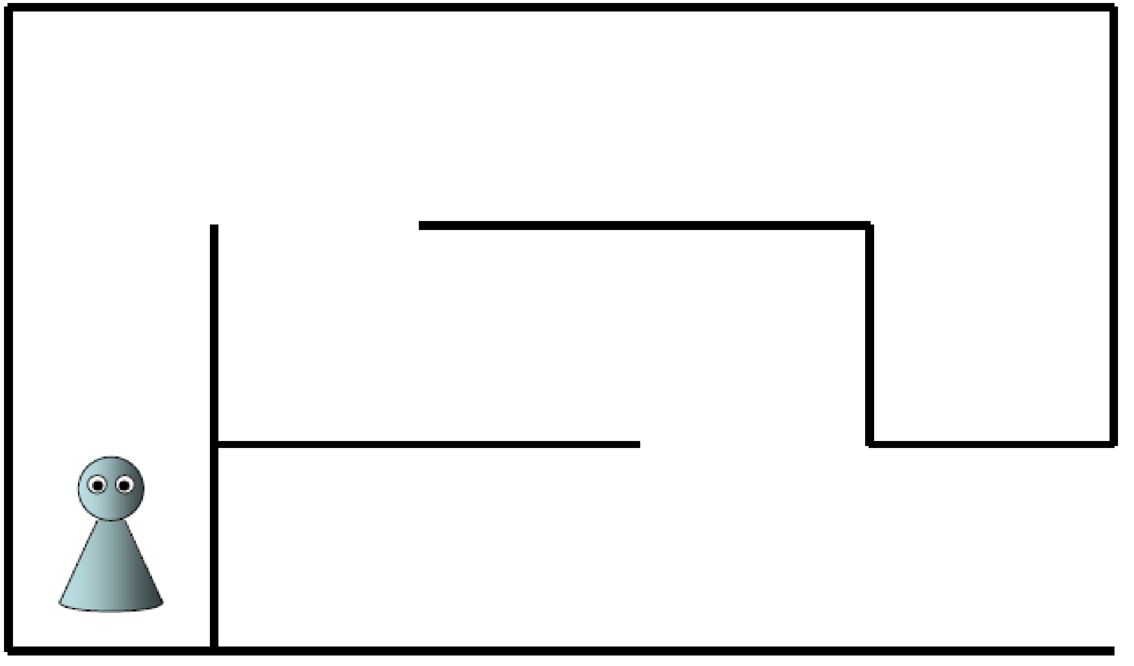


Example: An efficient routing algorithm lies at the heart of any network. But how to perform local routing, i.e., without global knowledge?

Recall the elephant analogy: in a distributed setting, a local node may only have a partial view of the network! How to do routing in this case?!

Example: Routing With A Local View

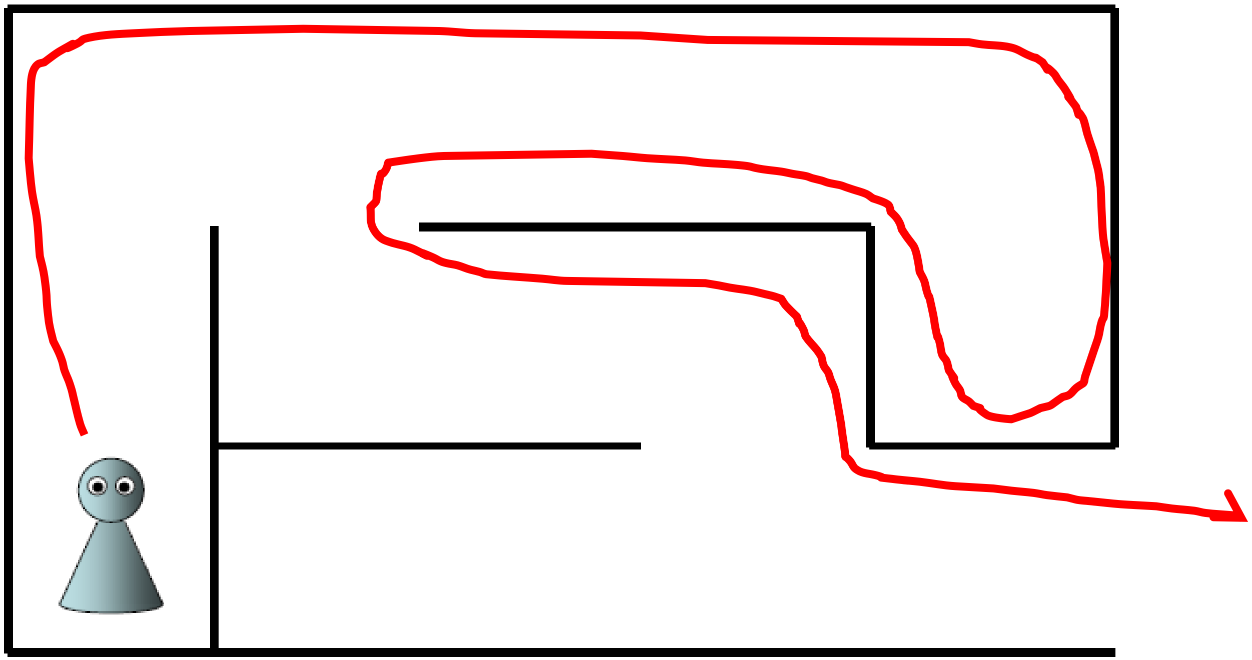
Example: How to get out of this maze in the dark?



It is dark and you do not have a torch!

Example: Routing With A Local View

Example: How to get out of this maze in the dark?

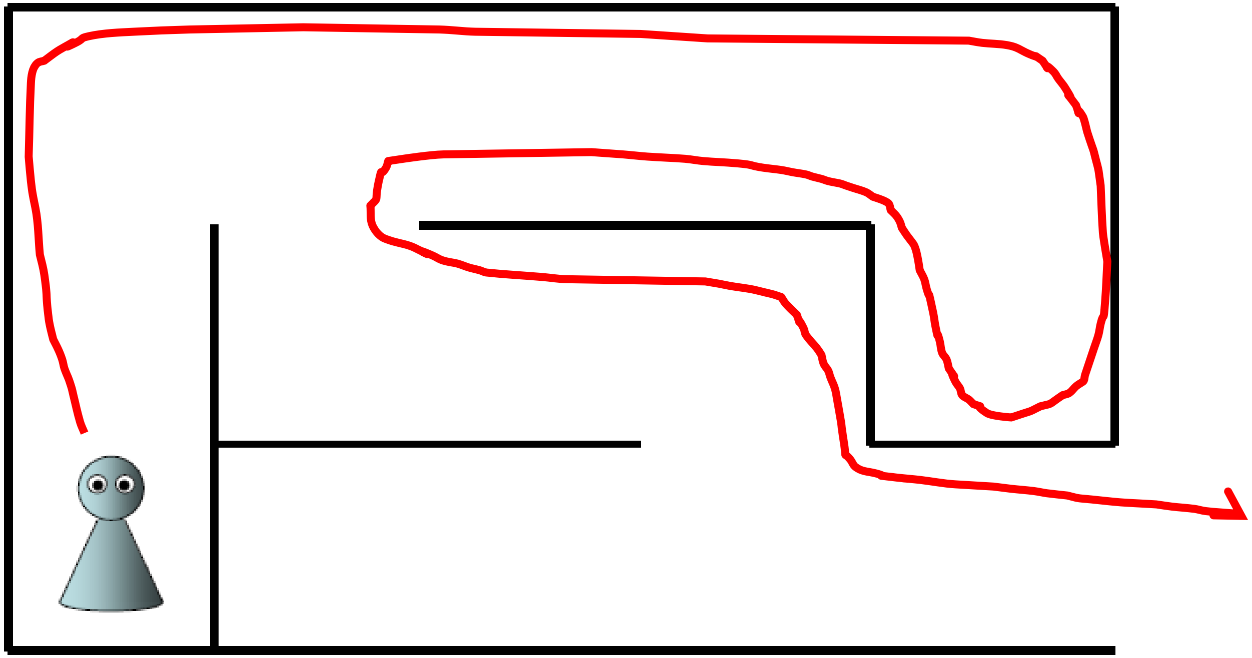


Famous trick:

Left-hand rule!

Example: Routing With A Local View

Example: How to get out of this maze in the dark?



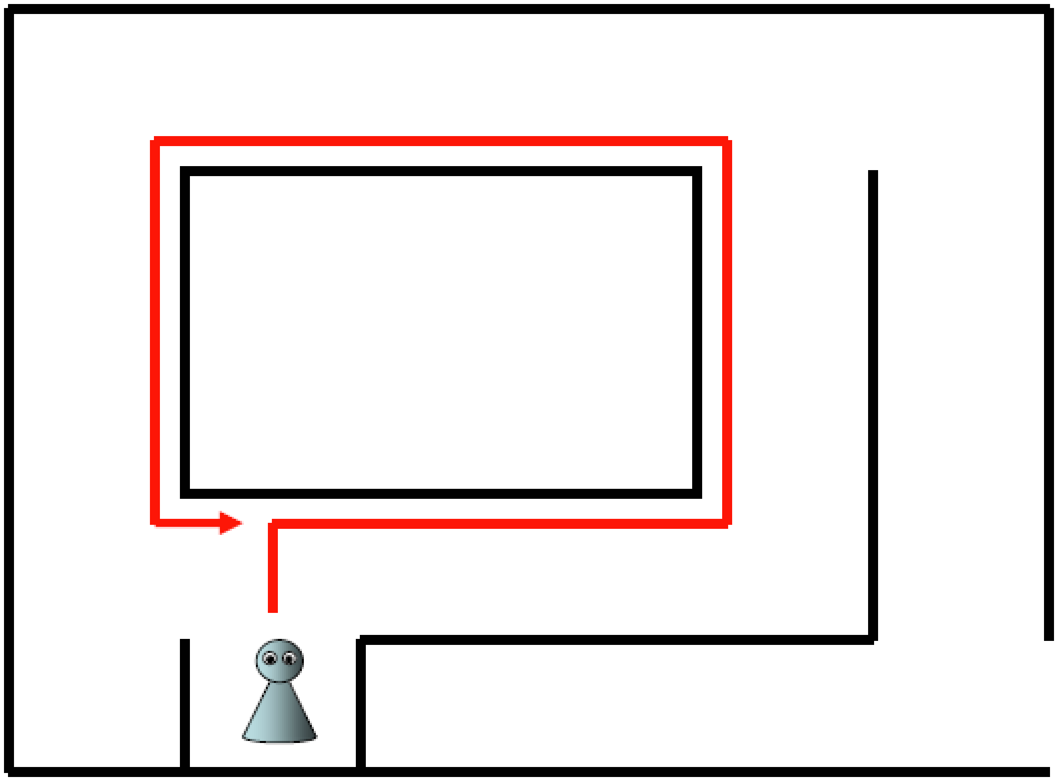
Nice: The algorithm does not need much memory!

Famous trick:

Left-hand rule!

Example: Routing With A Local View

Example: How to get out of the maze here?!

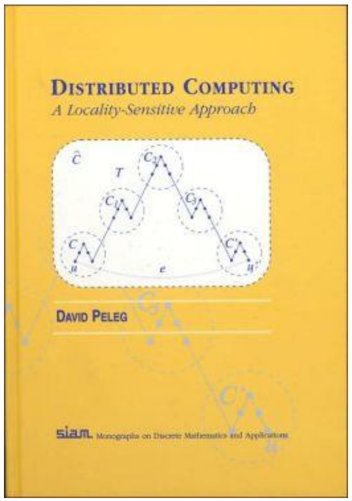


Argh......

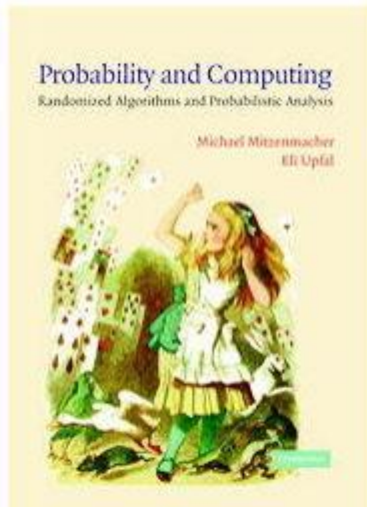
Homework: Local routing algorithm which can deal with mazes containing such a room?

Literature For This Course

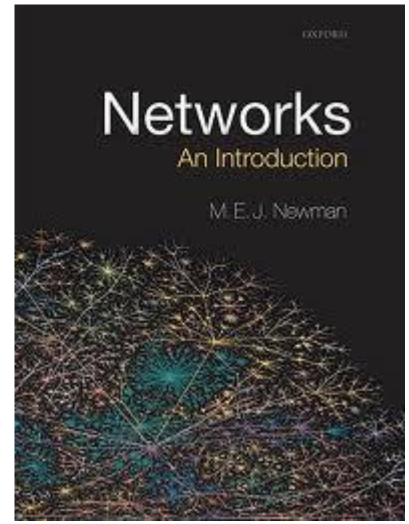
David Peleg “Distributed Computing”



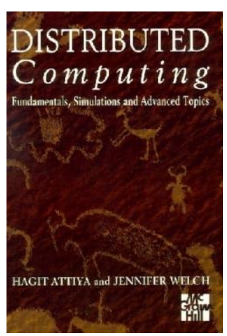
Upfal+Mitzenmacher “Probability and Computing”



Newman “Networks”



Attiya+Welch: “Distributed Computing”



+ Lecture Notes
+ Slides

Distributed Network Algorithms
(Lecture Notes for GIAN Course)

Assoc.-Prof. Dr. Stefan Schmid
Assoc-Prof. Dr. Partha Sarathi Mandal

Thanks to Prof. Dr. Roger Wattenhofer and
Prof. Dr. Christian Scheideler for basis of manuscript!

Summer 2016